

# The Case for the Balanced Instruction Set Computer

## 5T-7 - SIMP (単一命令流/多重命令パイプライン) 方式に向けた命令セット・アーキテクチャ

村上和彰 権 五鳳 富田眞治  
(九州大学)

### 1. はじめに

我々は、高速汎用プロセッサ・アーキテクチャとして、SIMP (Single Instruction stream / Multiple instruction Pipelining: 単一命令流/多重命令パイプライン) 方式を提案し、これに基づく試作プロセッサ『新風』/【Jmpu:】を開発している。<sup>1)</sup> また、SIMP方式に向けた命令セット (ISP: Instruction-Set Processor) アーキテクチャとして、BISC (Balanced Instruction Set Computer: 均衡命令セット・コンピュータ) を提案し、<sup>2)</sup> これを『新風』プロセッサにおいて具現化している。<sup>3)</sup>

本稿では、まずSIMP方式がISPアーキテクチャに求める要件を整理したあと、RISCおよびCISCと比較しながらBISCの特長について述べる。

### 2. ISPアーキテクチャの要件

SIMP方式自身はプロセッサ構成のための1論理方式であり、ISPアーキテクチャそのものではないし、また、ある特定のISPアーキテクチャを対象としたものでもない。さらに、SIMP方式では1台のプロセッサ内に複数本の命令パイプラインを備えるが、VLIW (Very Long Instruction Word: 超長形式機械命令) 方式とは異なり、命令パイプライン本数等のハードウェア構成をISPアーキテクチャに反映させる必要がない。このような点から、SIMP方式は通常の(単一)命令パイプライン方式同様、種々のISPアーキテクチャに対して適用可能な「汎用」のプロセッサ・アーキテクチャと言える。

しかし、SIMP方式は、実装した命令パイプライン本数に等しい複数個の命令を同時にフェッチ、デコードして実行を進めていくことから、唯一以下の制約をISPアーキテクチャに課すことになる；

- ① 単一命令長：命令長が固定かつ同一であること

この制約に適合しないISPアーキテクチャ(つまり、命令長が可変、あるいは、固定だが命令毎にサイズが異なる場合)では、命令のデコードが終了しないとその命令長が定まらない。これは、後続する命令との同時フェッチおよびデコードが不可能であることを意味しており、SIMP方式が適用できない(ただし、後述するように、機械命令レベルではなくマイクロ命令レベルでの適用は可能である)。

さらに、ISPアーキテクチャに以下の制約を加えた方が性能上好ましい；

- ② LOAD/STOREアーキテクチャ：レジスターレジスタ演算であること  
③ 固定サイクル演算：命令毎に決まった固定サイクルで演算が完了すること、すなわち、定型的演算であること  
④ 均衡演算時間：③に加えて、演算時間のバラツキが小さく均衡がとれていること

以上のうち、③④がSIMP方式に特徴的な要件であることから、これらを満たすISPアーキテクチャをBISC (Balanced Instruction Set Computer: 均衡命令セット・コンピュータ) と呼ぶ。このような要件が生じた背景は次節で述べる。

### 3. SIMP方式とISPアーキテクチャとの適合性

幅広いISPアーキテクチャのスペクトラムの中で、その端点に位置するRISCとCISC、および我々の提案するBISCに対してSIMP方式を適用する場合について考察する。

#### 3.1 RISC

RISCは、SIMP方式が極めて容易に適用できるISPアーキテクチャである。これは、RISCに対して一般に課せられている以下の(かなりきつい)制約によるものである；

- ① 単一命令長：通常のRISCでは単一の命令長(多くは32-bit)を採用している。これは、前述の通り、複数個の命令の同時フェッチ、デコードを可能とする。  
② LOAD/STOREアーキテクチャ：レジスターレジスタ演算は、命令間のデータ依存関係の検出を容易にする。すなわち、命令をデコードした段階でデータ依存関係の有無を判定することが可能である。これに対して、レジスターメモリ演算やメモリーメモリ演算では、メモリ・アドレスが生成されるまでデータ依存関係の検出が行えない。  
③ 単一サイクル演算：演算が1サイクルで完了することから、out-of-order実行制御(局所データフロー実行制御: データ依存関係のない命令から演算を開始する手法)を行なう際、命令のスケジューリングが容易に行える。

しかしながら、③の“単一”サイクル演算という制約は、命令パイプライン構成を単純にしRISCの命令数を大幅に減らす要因ではあるが、かなり厳しいものである。これにより、乗除算命令や浮動小数点演算命令などの基本命令でも、演算に複数サイクルを要することから、一般に提供されていない(提供されていてもコプロセッサで実行)。しかし、命令パイプラインに加えてその演算ステージにもパイプライン化を施すなら(演算パイ

プライン方式と呼ぶ), 複数サイクル演算命令でも1命令/サイクルのスループットを得ることが出来る。したがって, 演算パイプライン方式を採用する場合, ③の“単一”という制約はもはや本質的ではなく, これよりも緩い制約で充分である(すなわち, BISCの“固定”サイクル演算という制約)。

### 3.2 CISC

CISCにおいては, RISCに課せられているような制約が一切ないことから, SIMP方式の適用は困難である。特に, 命令長が一般に可変長もしくは複数固定長であるため, 複数命令の同時フェッチおよびデコードを不可能にしている。

しかしながら, CISCの機械命令の多くは一般にマイクロプログラム制御であるため, マイクロプログラムを実行するマイクロエンジンのレベルに対しては, SIMP方式を適用することが可能である。すなわち, マイクロ命令を単一命令長とすればよい。

### 3.3 BISC

BISCは, SIMP方式に最適なISPアーキテクチャとして提案しているものである。SIMP方式の長所を最大限に発揮するように, BISCには前述の通り以下の(RISCよりは緩い)制約が課せられる;

- ① 単一命令長: RISCと同様。
- ② LOAD/STOREアーキテクチャ: RISCと同様。
- ③ 固定サイクル演算: SIMP方式では, 命令の演算に対して並列演算パイプライン方式を適用する。演算パイプラインにおいてout-of-order実行制御を行なう場合, 命令毎に演算サイクルが固定されていることが命令スケジューリングのための必要条件であって, RISCのように必ずしも単一サイクルで演算が完了する必要はない。固定サイクル演算とは, 命令が演算パイプラインに入ったら一定のサイクル数で出て行くことを言う。これにより, RISCでは排除されている複数固定サイクル演算命令(例: 整数乗除算命令, 浮動小数点演算命令など)がBISCには含まれ得る。一方, CISCで一般的な可変長オペランド演算命令などは, その演算サイクル数が命令自身ではなくオペランドの長さや内容で決まる非定型的な可変サイクル演算である。可変サイクル演算では, 演算器を繰り返し使用し, しかも, その回数が実行してみないと判明しないことから, 後続命令の演算開始時点を決定するのが極めて困難である(演算パイプライン化自体も難しい)。したがって, このような可変サイクル演算命令はBISCには含まない。
- ④ 均衡演算時間: SIMP方式では正確な割込み(precise interrupt)を保証するため, 命令をフェッチした順序で命令実行を完了するようにしている。つまり, 同時にフェッチした複数個の命令(命令ブロックと呼ぶ)は, 同時に実行を完了しなければならない。したがって, 命令ブロック内で最も演算時間の長い命令によって, その命令ブロックの演算時間つまり命令パイプライン滞在時間が決まる。上記の③により多くの複数固定サイクル演算命令がBISCに含まれ得ること

表1. 演算時間の比較

プロセッサ命令	『新風』	SPARC MB86900	Motorola MC88100
整数加減算 論理演算	2	1	1
整数算除算	2 (乗算) (除算命令なし)	(乗除算命令 なし)	6 (乗算) (除算不明)
浮動小数点 演算	4~6 (加減乗算) (除算命令なし)	(コプロセッサ で実行)	5 (加減算) 6 (乗算) (除算不明)
ロード /ストア	4~6	2~4	3
分岐	4	1~2	(不明)
バラツキ	2~6	1~4	1~6

になるが, これらの演算時間に大きなバラツキがあると, 演算時間の短い命令を処理している命令パイプラインに無駄な遊びが生じることになる。結果的には, 多重命令パイプライン全体のスループットおよび応答速度の低下を招く。この弊害を防ぐには, 演算時間のバラツキがある範囲に収まるように命令を選択しなければならない。

### 4. 演算時間の均衡度

BISCでは, 演算時間のバラツキ範囲をどの程度まで抑えたらよいか重要となる。演算時間のバラツキと性能との相関関係については, 現在シミュレーションにより評価中である。

開発中の『新風』プロセッサの命令セットでは, 演算時間のバラツキを2~6マシン・サイクル(=1~3パイプライン・サイクル)の範囲に収めている。<sup>3)</sup>表1に他のプロセッサとの演算時間の比較を示す。

### 5. おわりに

以上, SIMP方式向けのISPアーキテクチャとして, BISCの特長を述べた。『新風』プロセッサのISPアーキテクチャはBISCの具体例であり, これについては参考文献3で報告する。

#### 参考文献

- 1) 村上ほか: 『新風』DTS-edition: SIMP (単一命令流/多重命令パイプライン)方式に基づくデスクトップ・スーパーコンピュータ, 情処37全大論文集, 4N-1 (1988年9月)
- 2) 村上ほか: SIMP (単一命令流/多重命令パイプライン)方式の構想, 情処研報, 88-CA-69-4 (1988年1月)
- 3) 久我ほか: 『新風』プロセッサの命令セット・アーキテクチャ, 本大会論文集 (1989年3月)